Understanding Semi Supervised Learning

Minor in AI - IIT ROPAR

4st April, 2025

Ava and the Map of the Unknown

Young explorer Ava dreamed of mapping the world. She had a few marked roads—clear places with names—and a vast, mysterious land full of clues but no labels.

At first, she had a guide telling her what each place was. That was **supervised learning**—accurate, but expensive.

Then she ventured alone, grouping places by look and feel. That was **unsupervised learning**—cheap, but vague.

One day, she found a tiny notebook with just a few labeled spots. She thought, "Can I use this to figure out the rest?"

And thus began her journey into **semi-supervised learning**.

She tried:

- **Self-training**: Trusting confident guesses
- **Co-training**: Working with her friend Ben, using different perspectives
- **Generative models**: Matching places to known patterns
- **Clustering**: Labeling similar groups together
- **Manifold learning**: Finding simple rules beneath complex places—like all dosas coming from the same batter

She even built a **Ladder Network**, recreating places layer by layer to truly understand them.

With time, Ava built a full map—not by labeling every point, but by smartly learning from the few she had.

That's the magic of semi-supervised learning.

What is Semi-Supervised Learning (SSL)?

Semi-Supervised Learning (SSL) is a powerful and pragmatic approach within machine learning that strategically combines a small amount of labeled data with a large pool of unlabeled data. In many real-world applications, acquiring labeled data is expensive, labor-intensive, and often requires domain expertise (such as radiologists for medical images, or linguists for language tasks). On the other hand, unlabeled data is plentiful and usually much easier to obtain—for example, raw texts from the web, untagged photos, or untranscribed speech.

SSL fills the space between two traditional learning paradigms: supervised learning and unsupervised learning. Supervised learning exclusively uses labeled examples, where each data point is paired with a corresponding label, such as an image annotated as "cat" or "dog." Unsupervised learning, in contrast, deals solely with unlabeled data and typically focuses on identifying structure within the data, such as clustering or dimensionality reduction. SSL, being a hybrid, harnesses the strengths of both—using the few labeled examples to guide the learning process while exploiting the structure inherent in the unlabeled data to enrich and reinforce the model's understanding.

The main objective of SSL is to achieve high performance without requiring large amounts of labeled data. By learning from both types of data, SSL models can uncover relationships, structures, and clusters that would remain hidden if only labeled data were used. This is especially crucial in domains where labeling is infeasible at scale. SSL models are designed to use unlabeled data not just as passive input but as a meaningful contributor to the training process, improving the generalization and robustness of the final model.

Motivation and Real-Life Examples

The practical motivation behind SSL is rooted in the imbalance between the availability of unlabeled and labeled data. In many domains, while there's no shortage of raw data, annotation is a bottleneck. To illustrate this, consider two concrete examples:

The first example is email spam detection. Suppose a user manually labels 100 emails as either "Spam" or "Not Spam." This is a small labeled dataset, as each labeling action consumes time and effort. Meanwhile, there may be an additional 10,000 emails available without any labels. A supervised model trained only on the 100 labeled samples would likely perform poorly due to limited data diversity. However, SSL makes it possible to improve model performance by also learning from the patterns, language structure, and metadata in the 10,000 unlabeled emails. Even without explicit labels, the model can identify recurring features common to spam messages and differentiate them from legitimate ones.

The second example comes from the medical field, where X-ray images must be reviewed and labeled by radiologists—a slow and expensive process. Suppose you have access to 500 labeled X-rays and 50,000 unlabeled ones. Instead of labeling all 50,000 (which might take months), SSL can help the model learn from the patterns present in both the labeled and unlabeled datasets. For instance, the model may notice shared structural patterns among images indicating pneumonia, even when labels are not provided. This enables the creation of a robust diagnostic tool with significantly reduced annotation costs.

Mathematical Formulation

To formally define Semi-Supervised Learning, let's assume we have two datasets. The first, denoted as

$$D_L = \{(x_i, y_i)\}_{i=1}^{l}$$

consists of l labeled samples where x_i represents the input features and y_i the corresponding label. The second dataset,

$$D_U = \{x_i\}_{i=l+1}^{l+u}$$

consists of u unlabeled samples. Together, the total dataset includes $l + u$ samples.

The learning task is to find a function $f(x)$—often a classifier or regression model—that generalizes well, not just over the labeled data but also in the context of the entire input space that includes the unlabeled data. This is usually done by minimizing a combined loss function that includes a supervised component (based on the labeled data) and an unsupervised component (based on patterns inferred from the unlabeled data). The inclusion of the unsupervised loss encourages the model to discover the structure in the data distribution, ensuring smoother decision boundaries and improved generalization.

Why SSL is Important

The significance of SSL lies in its ability to circumvent the need for large labeled datasets, which are often the main bottleneck in deploying machine learning systems. Labeling is expensive not just in monetary terms but also in the time and effort required from skilled professionals. For instance, in medical imaging, a single scan might take a specialist several minutes to evaluate, and massive datasets can require months of annotation.

Unlabeled data, by contrast, is ubiquitous. It exists in the form of logs, documents, emails, sensor readings, audio files, and more. These datasets are already collected in the course of normal business or operations, making them essentially "free" in terms of cost.

SSL unlocks the latent potential of these unlabeled datasets. It enables models to leverage both data types, reducing reliance on labels while improving performance. This is especially crucial in fields like:

Medical Imaging: Few labeled scans, massive hospital databases

Speech Recognition: Abundant audio, limited transcripts

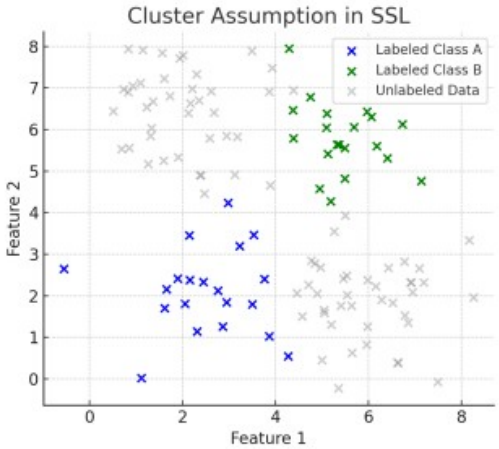
Text Classification: Rich corpora of raw documents

Autonomous Driving: Millions of road scenes, few annotated ones

Visualizing SSL

To visualize how SSL works, imagine a 2D space where data points are plotted based on their features. Labeled points might be red (class A) and blue (class B). The labeled data, being sparse, provide only a rough idea of the boundary between classes. Unlabeled points, though colorless, populate the space and form discernible clusters and patterns.

A model trained only on labeled points might draw a poor boundary, cutting across dense regions. SSL techniques use the unlabeled points to infer that the boundary should avoid cutting through dense clusters and instead run through sparse regions, aligning better with the true data distribution. This helps the model achieve low-density separation, leading to improved classification accuracy.



Comparing Learning Paradigms

SSL stands out when compared to traditional learning paradigms. In supervised learning, only labeled data is used, which limits scalability and adaptability when labeled data is scarce. In unsupervised learning, there is no label information at all, which restricts tasks to clustering or representation learning.

SSL, by utilizing a mix of few labeled and many unlabeled examples, offers a compromise. It provides the model with direct supervision where available and allows it to generalize patterns from the unlabeled data. This is particularly useful for tasks like spam detection, where collecting a few labeled samples is feasible, but labeling thousands is not.

Learning Type	Data Used	Example
Supervised	Only labeled	Image classification
Unsupervised	Only unlabeled	Customer segmentation
Semi-Supervised	Few labeled + many unlabeled	Spam detection with limited labels

Key Assumptions in SSL

Self-Training

Self-training assumes that if a model is confident in its prediction, that prediction is probably correct. The process begins by training the model on labeled data. It then predicts labels for the unlabeled set. If certain predictions exceed a confidence threshold (e.g., 98% sure the item is an "Apple"), those pseudo-labeled instances are added to the training set. The model is then retrained with this expanded dataset. This cycle continues, with the model progressively becoming more robust. However, caution is needed: if the model confidently makes wrong predictions, it may reinforce errors.

Co-Training

Co-training assumes the data can be represented from two or more independent and sufficient views. Two separate models are trained on different feature subsets. For example, when classifying web pages:

- View 1: Page content (words like "goal", "score")
- View 2: Hyperlinks (e.g., links to ESPN)

Each classifier labels unlabeled examples for the other. This mutual learning process helps both models improve as long as the views are conditionally independent and each view alone is sufficient for classification.

Generative Model Assumption

This assumption is that the data are generated by underlying probabilistic distributions (e.g., Gaussians). If the model can fit a probability distribution to the labeled and unlabeled data (say, one Gaussian per class), it can assign a new point to the most likely distribution (i.e., class). This method is effective when the actual data-generating process aligns with the assumed distributions.

Cluster Assumption

This states that points in the same cluster likely share the same label. By clustering both labeled and unlabeled data (using, for instance, K-means), we can assign cluster-wide labels based on the few labeled samples.

Low-Density Separation

The principle here is that a good classifier should place its decision boundary in low-density regions—areas of the feature space with fewer data points. This reduces the chance of misclassifying similar samples. This is the rationale behind SSL techniques like Transductive SVMs.

Manifold Assumption

This powerful idea proposes that high-dimensional data often lie on a low-dimensional manifold. For example, images of handwritten digits live in a 784-dimensional space (28×28 pixels), but the variation is smooth and controlled (stroke, slant, thickness), forming a lower-dimensional structure. SSL models can propagate labels along this manifold, assigning similar labels to nearby points.

Summary Table of SSL Assumptions

Assumption	Key Idea	Example
Self-Training	Confident predictions are likely correct	Red fruit labeled as "Apple"
Co-Training	Independent views teach each other	Text + hyperlink views of a webpage
Generative Model	Data comes from known distributions	Gaussian clusters in 2D
Cluster	Same cluster → same label	Cat/dog image clusters
Low-Density	Boundaries avoid dense regions	Two moons toy dataset
Manifold	Labels vary smoothly along manifolds	Handwriting of digit "3"

Related Paradigms to SSL

Transfer Learning

Transfer Learning involves transferring knowledge learned in one domain (called the source domain) to a different but related domain (the target domain). Usually, this involves pretraining a model on a large labeled dataset (e.g., ImageNet), and then fine-tuning it on a smaller labeled dataset from the target domain (e.g., X-ray classification). The primary goal is to reuse features and model weights learned from one context to improve performance in another, especially when labeled data is limited in the target domain.

Weakly-Supervised Learning

Weakly-Supervised Learning focuses on using imperfect labels, rather than few labels. These imperfections might include:

- Noisy labels (e.g., auto-tagged tweets)
- Incompletely labeled data (e.g., videos labeled only by title)
- Coarse labels (e.g., document labeled with topics but no sentence-level tags)

While SSL assumes you have a few high-quality labels, Weak Supervision tolerates many low-quality labels. The model then learns to identify signal amidst the noise. SSL and Weak Supervision can be combined for greater flexibility in real-world tasks.

Positive and Unlabeled (PU) Learning

PU Learning is a special case of SSL where only positive examples are labeled, and the rest are unlabeled. There are no known negative examples. A common use case is spam detection, where spam emails are flagged, but non-spam (ham) emails are not explicitly labeled.

Meta-Learning

Meta-Learning, or "learning to learn," aims to enable a model to adapt quickly to new tasks with minimal data. Unlike SSL, which works on a single large task, Meta-Learning trains across many small tasks—for instance, classifying new classes using only 1 or 5 labeled examples per class.

Self-Supervised Learning

Self-Supervised Learning removes external labels entirely. Instead, it creates pseudo-labels or proxy tasks (called pretext tasks) directly from the data. Examples include predicting missing image patches (in models like SimCLR), predicting the next word (BERT), or reconstructing masked tokens.

Summary Table: Related Paradigms

Paradigm	Label Setup	Typical Use Case
Semi-Supervised	Few labeled + many unlabeled	Text/image classification
Transfer Learning	Pretrained model + few new labels	Domain adaptation
Weak Supervision	Many noisy/incomplete labels	Learning from web-labeled datasets
PU Learning	Only positive + unlabeled	Fraud or spam detection
Meta-Learning	Few-shot tasks across tasks	Few-shot classification
Self-Supervised	No labels at all	Feature extraction

Inductive vs Transductive Learning in SSL

Inductive SSL

In Inductive Learning, the goal is to learn a general function $f(x)$ that can be applied to any new, unseen input. This approach trains the model on the labeled and unlabeled data, often using methods like pseudo-labeling, consistency regularization, or data augmentation.

Transductive SSL

In Transductive Learning, the model's goal is not to learn a general rule but to label only the unlabeled data provided during training. This is more akin to propagating known labels to a fixed batch of unlabeled data.

Comparison Table

Feature	Inductive	Transductive
Goal	Learn a general classifier $f(x)$	Predict labels for a fixed unlabeled set
Output	Deployable model	Only predictions, no reusable model
Generalization	Works on unseen data	Cannot generalize to new data
Examples	Pseudo-labeling, MixMatch	Label propagation, TSVM
Advantage	Real-world deployment possible	High accuracy on current batch
Limitation	May sacrifice some accuracy	Not usable on future data

Ladder Networks and Π-Models (Pi-Models)

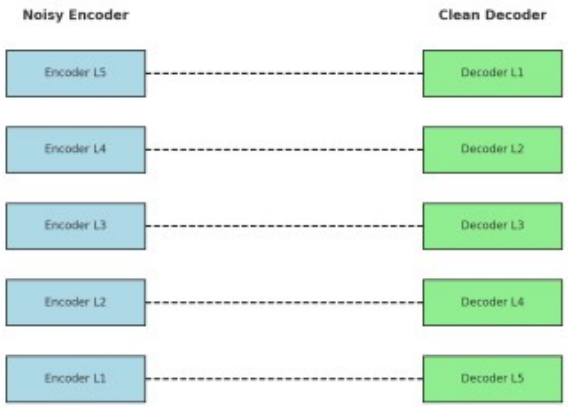
Ladder Networks

A Ladder Network is a deep neural network architecture that combines supervised and unsupervised learning by denoising internal representations. It consists of:

- A bottom-up encoder: adds noise to the input and passes it through the network
- A top-down decoder: reconstructs the clean version of each layer's activation
- Skip connections between encoder and decoder layers, forming a "ladder" shape

Each layer in the encoder learns noisy representations, while the decoder learns to recover the clean activations. Supervised loss is applied at the top (final output), and unsupervised loss is applied to each hidden layer. The total loss function is a combination of:

- Supervised loss (e.g., cross-entropy for classification)
- Layer-wise reconstruction loss, each weighted by a parameter λ_l



Π-Model (Pi-Model)

The Π-Model introduces consistency regularization. It is based on the idea that a model's prediction should remain stable under small perturbations of the input.

Mechanism:

- The same input x is passed twice through the same neural network with different augmentations or dropout noise.
- This results in two outputs: $f_1(x + \epsilon_1)$ and $f_2(x + \epsilon_2)$
- The model is trained to make these outputs match:

$$L_{unsup} = \|f_1(x) - f_2(x)\|^2$$

Comparison: Ladder vs Π-Model

Aspect	Ladder Network	Π-Model
Core Idea	Denoising hidden representations	Consistency under perturbations
Architecture	Encoder + Decoder	Single model, run twice
Unsupervised Loss	Reconstruction loss per layer	Consistency (MSE) loss on output
Noise Type	Gaussian noise at each layer	Gaussian/dropout noise at input

Key Takeaways

- **Semi-Supervised Learning (SSL)** combines few labeled samples with many unlabeled ones to improve model performance.
- It is effective in reducing labeling effort while still achieving high accuracy.
- SSL is valuable in areas with expensive labeling like medical imaging, speech, and text processing.
- Assumes that data has structure: clusters, low-density regions, or low-dimensional manifolds.
- Popular approaches include self-training, co-training, generative models, clustering, and manifold learning.
- Advanced models like **Ladder Networks** and Π-Models use denoising and consistency to enhance learning.
- Related paradigms are Transfer Learning, Weak Supervision, PU Learning, Meta-Learning, and Self-Supervised Learning.
- SSL works in both **inductive** (generalizing to new data) and **transductive** (labeling current data only) settings.